International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 11
ISSN: 2454-4248
79 – 85

Survey on Faster Region Convolution Neural Network for Object Detection

Mrs. Swetha M S[1]
Assistant Professor, Department of IS&E, BMS Institute of Technology & Management, Yelahanka, Bangalore - 560064, Karnataka,
E-mail: swethams_ise2014@bmsit.in

Ms. Srishti Suman[2]
Department of IS&E, BMS Institute of Technology & Management, Yelahanka, Bangalore -560064, Karnataka,
E-mail: srishtibeg@gmail.com

Mr. Muneshwara M S[3]
Assistant Professor, Department of CSE, BMS Institute of Technology & Management, Yelahanka, Bangalore - 560064,
E-mail:muneshwarams@bmsit.in

Dr. Thungamani M[4]
Associate Professor, Department of CSE,
Assistant Professor, Dept. of CSE, COH UHS
Karnataka,Campus GKVK BANGALORE 560065,
E-mail: thungamani_k@rediffmail.com

Abstract: Convolution Neural Networks uses the concepts of deep learning and becomes the golden standard for image classification. This algorithm was implemented even in complicated sights with multiple overlapping objects, different backgrounds and it also successfully identified and classified objects along with their boundaries, differences and relations to one another. Then comes Region-based Convolutional Neural Networks(R-CNN)which is further more described into two types that is Fast R-CNN and Faster R-CNN. This R-CNN method is to use selective search to extract only 2000 regions from the image and cannot be implemented in real time as it would take 47 sec approximately for each test image. Then comes the fast R-CNNin which changes are made to overcome the drawbacks in R-CNN algorithm in which the 2000 region proposals are not fed to the CNN instead the image is fed directly to the CNN to generate Convolutional feature map. This was then replaced by faster R-CNN which came up with an object detection algorithm that eliminates the selective search algorithm to perform the operation. This algorithm takes 0.2 sec approximately for the test image and we will be using this for real time object detection.So, basically in this paper we are doing research on Faster R-CNN that is being used for object detection method.

Keywords: *Convolution Neural Networks (CNN), Region-based Convolutional Neural Networks(R-CNN),Fast R-CNN,fasterR-CNN*

*****

I. INTRODUCTION

In recent years, with the rapid development of machine learning and deep learning, a number of research areas with respect to these concepts has increased immensely. Along with this there is a continuous improvement of convolution neural networks(CNN). It efficiently improves applications such as face recognition, object detection, object tracking, object relationship etc. Convolution Neural Network is efficiently improving applications such as face recognition, object detection, object tracking, object relationship etc. Object detection is one of the most important applications in the field of deep learning and image processing, and has been the focus of research.

Convolution neural network has made great progress in object detection. Object detection has therefore developed from the single object recognition to the multi-object recognition to real-time object recognition. Since the R-CNN was proposed in 2014, which is based on deep convolutional neural networks, it has developed greatly. Subsequently, lots of improved methods based on the R-CNN, such as Spp-net, fast R-CNN, faster RCNN and R-FCN , emerged in the object detection area. These methods achieved high accuracies, but their network structures are relatively complex. The goal of R-CNN is to take in an image, and correctly identify where the main objects (via a bounding box) in the image. The inputs are images and the outputs will be bounding boxes and the labels for each objects in the images. R-CNNproposes a bunch of boxes in the image and see if any of them actually correspond to an object.R-CNN creates these bounding boxes, or region proposals, using a process called Selective Search . At a high level, Selective Search looks at the image through windows of different sizes, and for each size tries to group together adjacent pixels by texture, color, or intensity to identify objects.R-CNN warps the region to a standard square size and passes it through to a modified version of AlexNet.On the final layer of the CNN, R-CNN adds a Support Vector Machine (SVM) that simply classifies whether this is an object and if it is an object then what is that object. The steps included in R-CNN are:

1. Generate a set of proposals for bounding boxes.
2. Run the images in the bounding boxes through a pre-trained AlexNet and finally an SVM to see what object the image in the box is.

 ISSN: 2454-4248
79 – 85

R-CNN works really well, but is really quite slow as it requires a forward pass of the CNN (AlexNet) for every single region proposal for every single image.

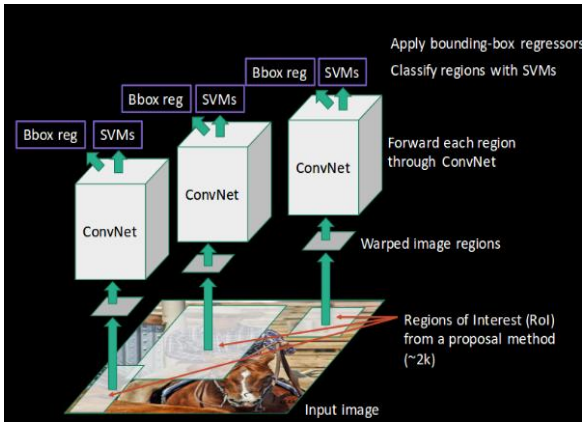

Fig 1: R-CNN Model

R-CNN author solved some of the drawbacks of R-CNN to build a faster object detection algorithm and was called Fast R-CNNandis similar to the R-CNN algorithm. But, instead of feeding the region proposals to the CNN, we feed the input image to the CNN to generate a convolutional feature map.Fast R-CNN is significantly faster in training and testing sessions over When you look at the performance of Fast R-CNN during testing time, including region proposals slows down the algorithm significantly when compared to not using region proposals. Therefore, region proposals become bottlenecks in Fast R-CNN algorithm affecting its performance.

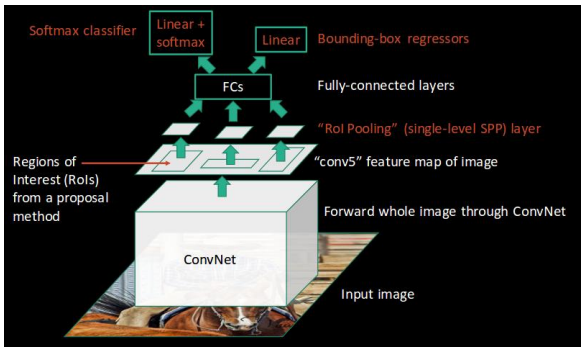

Fig 2: Fast R-CNN

Both of the above algorithms(R-CNN & Fast R-CNN) uses selective search to find out the region proposals. Selective search is a slow and time-consuming process affecting the performance of the network. Therefore Faster R-CNN is a model which has come up with the object detection algorithm that eliminates the selective search algorithm and lets the network learn the region proposals. Here,the image is provided as an input to a convolutional network which provides a convolutional feature map. Instead of using selective search algorithm on the feature map to identify the region proposals, a separate network is used to predict the region proposals.The predicted region proposals are then reshaped using a RoI pooling layer which is then used to classify the image within the proposed region and predict the offset values for the bounding boxes.

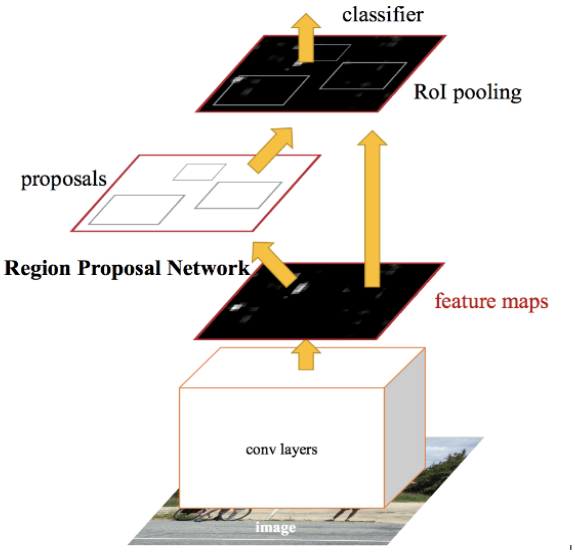

Fig 3: Faster R-CNN

II. LITERATURE SURVEY

2.1 Object-Oriented Image Analysis Using the CNN Universal Machine

In recent years, great efforts have been devoted to the study of coding techniques providing high compression ratios, while maintaining good picture quality. Among these techniques, the object-oriented coding approach represents an interesting and sophisticated method for video coding.

2.2 Decomposition of the Nodal Conductance Matrix of a Planar Resistive Grid and Derivation of Its Eigenvalues and Eigenvectors Using the Kroenke Product and Sum with Application.

This paper proposesresistive grids that have long been used in spatial and spatiotemporal filtering. It is also well known that the linear and planar resistive grids can be cast as 1-D and 2-D low-pass Cellular Neural Network (CNN) image filters, respectively, by introducing a unity capacitor between each node and ground.

2.3 Fault-Tolerant Design of Analogic CNN Templates and AlgorithmsÐPart I: The Binary Output Case

This paper proposescellular neural/nonlinear networks (CNN's)and the CNN universal machine (CNN-UM) were invented in1988 and 1992. At that time, due to the lack of any programmable analogic CNN chips, the templates were designed to be operational on ideal CNN structures. These structures were simulated on digital computers. During this

 ISSN: 2454-4248
79 – 85

period, and later, several template learning and optimization methods were developed. The goal of these methods was template generation, dealing mainly with ideal CNN behavior and without much regard to robustness issues. As a result, a large number of templates were introduced. Some of them were designed by using template learning algorithms, but most of them were created relying on ad hoc methods and intuition. A representative collection of these templates can be found in the CNN Software Library.

2.4 VPRS-Based Regional Decision Fusion of CNN and MRF Classifications for Very Fine Resolution Remotely Sensed Images

This paper proposesremote sensing technologies have evolved greatly since the launch of the first satellite sensors, with a significant change being the wide suite of very fine spatial resolution (VFSR) sensors borne by diverse platforms (satellite, manned aircraft, or unmanned aerial vehicles).

2.5 Feature Extraction for Classification of Hyperspectral and LiDAR Data Using Patch-to-Patch CNN.

This paper proposes ENSOR technology has experienced important **a**dvances lately, allowing us to measure variousmanuscript. This work was supported in part by the National Natural Science Foundation of China under Grant NSFC-91638201 and Grant 61571033, in part by the Beijing Natural Science Foundation under Grant 4172043, in part by the Beijing Nova Program under Grant Z171100001117050, and in part by the Fundamental Research Funds for the Central Universities under Grant BUCTRC201615.

2.6 A CNN-Based Defect Inspection Method for Catenary Split Pins in High-Speed Railway

This paper is about the railway network,as it is becoming heavy, the long-term impact and vibration triggered by vehicle operation could make fasteners of catenary support devices (CSDs), especially split pins (SPs), severely loose or missing. SPs play an important role in fixing joint components on CSDs whose function is mainly to maintain the conductor height and stagger of the contact line. The SPs loose or fall off would cause the structure of CSDs unstable, or even may cut off the electric power transmission from the contact line to vehicle.

2.7 Real-Time Taxi-Passenger Prediction with L-CNN

This paper proposes transportation planning and solutions have an enormous impact on city life. To minimize the transport duration, transportation service providers, such as Didi and Uber, should understand and elaborate the mobility of a city, which becoming more and more likely to be achieved with the popularity of Internet of Vehicle.

2.8 Fast R-CNN

This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network 9× faster than R-CNN obtained from a stationary camera. The developed vehicle detection and counting algorithm was implemented and tested also on an embedded platform of smart cameras.

2.9 Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

This paper proposesrecent advances in object detection are driven by the success of region proposal methods and region-based convolutional neural networks (R-CNNs). Although region-based CNNs were com-putationally expensive as originally developed in their cost has been drastically reduced thanks tosharing convolutions across proposals. The latest incarnation, Fast R-CNNachievesnear real-time rates using very deep networks, when ignoring the time spent on region proposals. Now, proposals are the test-time computational bottleneck in state-of-the-art detection systems.

III. PROPOSED SYSTEM

The proposed system involves the optimization of Our object detection system, called Faster R-CNN which is composed of two modules.The first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector that uses the proposed regions.The entire system is a single, unified network for object detection.Using the recently popular terminology of neural networks with mechanisms, the RPN module tells the Fast R-CNN module where to look.A Region Proposal Network (RPN) takes an image (of any size) as input and outputs a set of rectangular object proposals, each with an abjectness score.

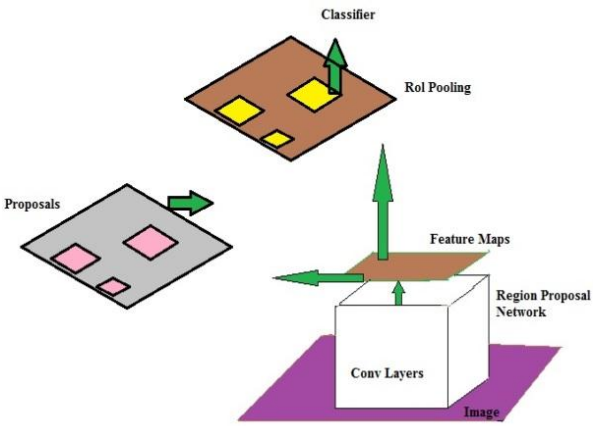

Fig 4: Faster R-CNN, single unified network model

 ISSN: 2454-4248
79 – 85

The steps followed by a Faster R-CNN algorithm to detect objects in an image:

1. Take an input image and pass it to the ConvNet which returns feature maps for the image
2. Apply Region Proposal Network (RPN) on these feature maps and get object proposals
3. Apply ROI pooling layer to bring down all the proposals to the same size
4. Finally, pass these proposals to a fully connected layer in order to classify any predict the bounding boxes for the image.Working of the Faster R-CNN:

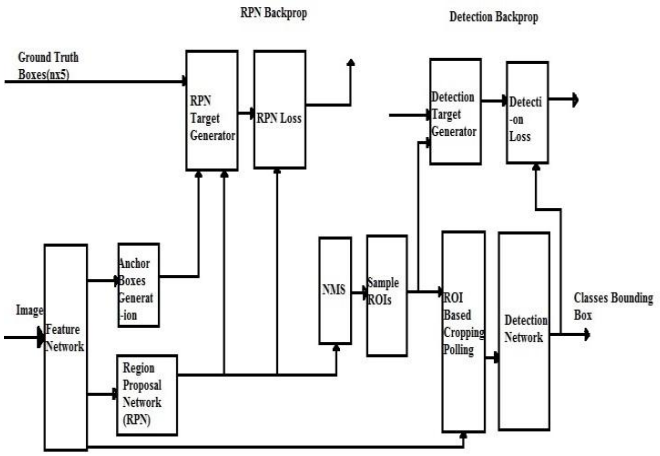

Fig 5: faster R-CNN working

The implementation of the Faster R-CNN involves the following steps:

1. **Train_images:** Images that we will be using to train the model. We have the classes and the actual bounding boxes for each class in this folder.
2. **Test_images:** Images in this folder will be used to make predictions using the trained model. This set is missing the classes and the bounding boxes for these classes.
3. **Train.csv:** Contains the name, class and bounding box coordinates for each image. There can be multiple rows for one image as a single image can have more than one object.

We train and test both region proposal and object detection networks on images of a single scaleNow, we need to move the train_ images and test images folder, as well as the train.csv file, to the cloned repository. In order to train the model on a new dataset, the format of the input should be: filepath, x1, y1, x2, y2,class_namewhere,

- file path is the path of the training image
- x1 is the xmin coordinate for bounding box
- y1 is the ymin coordinate for bounding box
- x2 is the xmax coordinate for bounding box
- y2 is the ymax coordinate for bounding box
- class_name is the name of the class in that bounding box

Multi-scale feature extraction (using an image pyramid) may improve accuracy but does not exhibit a good speed-accuracy trade-off.After this we need to train our model as explained above.

IV. METHODOLOGY

4.1 RPN

A Region Proposal Network (RPN) takes an image (of any size) as input and outputs a set of rectangular object proposals, each with an objectness score.To generate region proposals, we slide a small network over the convolutional feature map output by the last shared convolutional layer.This small network takes as input an n × n spatial window of the input convolutional feature map. Each sliding window is mapped to a lower-dimensional feature.This feature is fed into two sibling fullyconnected layers—a box-regression layer (reg) and a box-classification layer (cls).This architecture is naturally implemented with an n×n convolutional layer followed by two sibling 1 × 1 convolutional layers.

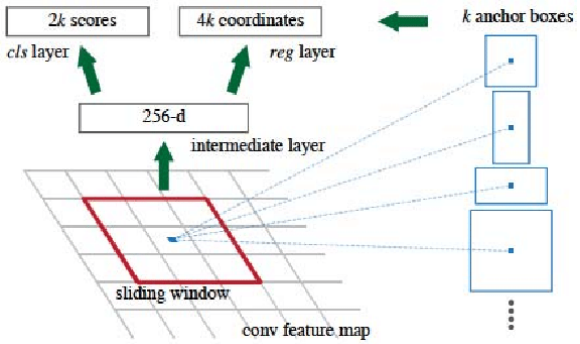

Fig 6:Region Proposal Network (RPN)

4.1.1 Anchors

At each sliding-window location, we simultaneously predict multiple region proposals, where the number of maximum possible proposals for each location is denoted as 'k'. So the reg layer has 4k outputs encoding the coordinates of k boxes, and the cls layer outputs 2k scores that estimate probability of object or not object for each proposal4 . The k proposals are parameterized relative to k reference boxes, which we callanchors. An anchor is centered at the sliding window in question, and is associated with a scale and aspect ratio.By default we use 3 scales and 3 aspect ratios, yielding k = 9 anchors at each sliding position. For a convolutional feature map of a size W × H , there are W Hk anchors in total.

TYPES OF ANCHORS:

- Translation Invarient Anchors
- Multiscale Anchors

 ISSN: 2454-4248
79 – 85

4.1.1.1Translation-Invariant Anchors

An important property of our approach is that it is translation invariant, both in terms of the anchors and the functions that compute proposals relative to the anchors. If one translates an object in an image, the proposal should translate and the same function should be able to predict the proposal in either location. This translation-invariant property is guaranteed by our method.The translation-invariant property also reduces the model size. MultiBox has a (4 + 1) × 800-dimensional fully-connected output layer, whereas our method has a (4 + 2) × 9-dimensional convolutional output layer in the case of k = 9 anchors. As a result, our output layer has 2.8 × 104 parameters.We expect our method to have less risk of overfitting on small datasets, like PASCAL VOC.

4.1.1.2 Multi-Scale Anchors as Regression References

Our design of anchors presents a novel scheme for addressing multiple scales .,There have been two popular ways for multi-scale predictions. The first way is based on image/feature pyramids, e.g., in DPM and CNNbased methods. The images are resized at multiple scales, and feature maps HOGor deep convolutional features are computed for each scale This way is often useful but is time-consuming. The second way is to use sliding windows of multiple scales (and/or aspect ratios) on the feature maps. For example, in DPM , models of different aspect ratios are trained separately using different filter sizes. Our method classifies and regresses bounding boxes with reference to anchor boxes of multiple scales and aspect ratios. It only relies on images and feature maps of a single scale, and uses filters (sliding windows on the feature map) of a single size.

4.1.2 Loss Function

For training RPNs, we assign a binary class label (of being an object or not) to each anchor. We assign a positive label to two kinds of anchors: (i) the anchor/anchors with the highest Intersection-overUnion (IoU) overlap with a ground-truth box, or (ii) an anchor that has an IoU overlap higher than 0.7 with 5 any ground-truth box. Note that a single ground-truth box may assign positive labels to multiple anchors. Usually the second condition is sufficient to determine the positive samples; but we still adopt the first condition for the reason that in some rare cases the second condition may find no positive sample. We assign a negative label to a non-positive anchor if its IoU ratio is lower than 0.3 for all ground-truth boxes. Anchors that are neither positive nor negative do not contribute to the training objective.

4.1.3Training the RPN

For training the RPN, first a number of bounding boxes are generated by a mechanism called anchor boxes. Every ‘pixel’ of the feature image is considered an anchor. Each anchor corresponds to a larger set of squares of pixel in the original image some reshaping is usually done on the original image before feature extraction. All the anchors are positioned uniformly across both dimensions of the (reshaped) image. The input that is required from the feature generation layer to generate anchor boxes is the shape of the tensor, not the full feature tensor itself.A number of rectangular boxes of different shapes and sizes are generated centered on each anchor. Usually 9 boxes are generated per anchor (3 sizes x 3 shapes).

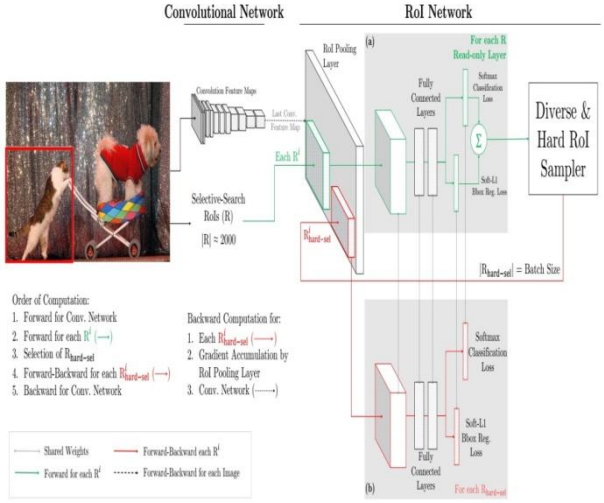

Fig 7: Training the RPN

The RPN can be trained end-to-end by backpropagation and stochastic gradient descent (SGD) [35]. We follow the “image-centric” sampling strategy from [2] to train this network. Each mini-batch arises from a single image that contains many positive and negative example anchors. It is possible to optimize for the loss functions of all anchors, but this will bias towards negative samples as they are dominate.We randomly initialize all new layers by drawing weights from a zero-mean Gaussian distribution with standard deviation 0.01. All other layers (i.e., the shared convolutional layers) are initialized by pretraining a model for ImageNet classification.

4.2 Sharing Features for RPN and Fast R-CNN Thus far we have described how to train a network for region proposal generation, without considering the region-based object detection CNN that will utilize these proposals. For the detection network, we adopt Fast R-CNN . Next we have learnt the algorithms that a unified network is composed of RPN and Fast R-CNN with shared convolutional layers . Both RPN and Fast R-CNN, trained independently, and have modified their convolutional layers in different ways. We therefore need to develop a technique that allows for sharing convolutional layers between the two networks, rather than learning two separate networks. We discuss three ways for training networks with features shared:

 ISSN: 2454-4248
79 – 85

(i) Alternating training: In this solution, we first train RPN, and use the proposals to train Fast R-CNN. The network tuned by Fast R-CNN is then used to initialize RPN, and this process is iterated. This is the solution that is used in all experiments in this paper.

(ii) Approximate joint training: In this solution, the RPN and Fast R-CNN networks are merged into one network during training .In each SGD iteration, the forward pass generates region proposals which are treated just like fixed, pre-computed proposals when training a Fast R-CNN detector. The backward propagation takes place as usual, where for the shared layers the backward propagated signals from both the RPN loss and the Fast R-CNN loss are combined. This solution is easy to implement. But this solution ignores the derivative w.r.t. the proposal boxes' coordinates that are also network responses, so is approximate.

(iii) Non-approximate joint training: As discussed above, the bounding boxes predicted by RPN are also functions of the input. The RoI pooling layer in Fast R-CNN accepts the convolutional features and also the predicted bounding boxes as input, so a theoretically valid backpropagation solver should also involve gradients w.r.t. the box coordinates. These gradients are ignored in the above approximate joint training. In a non-approximate joint training solution, we need anRoI pooling layer that is differentiable w.r.t. the box coordinates.

4-Step Alternating Training: In this paper, we adopt a pragmatic 4-step training algorithm to learn shared features via alternating optimization.

STEP 1: we train the RPN as described in Section 5.1.3. This network is initialized with an ImageNet-pre-trained model and fine-tuned end-to-end for the region proposal task.

STEP 2: we train a separate detection network by Fast R-CNN using the proposals generated by the step-1 RPN. This detection network is also initialized by the ImageNet-pre-trained model. At this point the two networks do not share convolutional layers.

STEP 3: we use the detector network to initialize RPN training, but we fix the shared convolutional layers and only fine-tune the layers unique to RPN. Now the two networks share convolutional layers.

STEP 4: keeping the shared convolutional layers fixed, we fine-tune the unique layers of Fast R-CNN.

As such, both networks share the same convolutional layers and form a unified network. A similar alternating training can be run for more iteration, but we have observed negligible improvements.

V. CONCLUSION

In this paper we have presented CNN, RCNN, and FAST R-CNN and focused properly on Faster R-CNN network. Here we have used RPNs for efficient and accurate region proposal generation. By sharing convolutionalfeatures with the down-stream detection network, the region proposal step is nearly cost-free. Our method enables a unified, deep-learning-based object detection system to run at near real-time frame rates. The learned RPN also improves region proposal quality and thus the overall object detection accuracy.

REFERENCES

[1] Ross Girshick," Fast R-CNN",International Conference onComputer Vision, IEEE, 2015.

[2] Shaoqing Ren, Kaiming He Ross Girshick Jian Sun "Faster R-CNN: Towards Real-Time Object Detectionwith Region Proposal Networks"

[3] Giuseppe Grassi , Luigi Alfredo Grieco" Object-Oriented Image Analysis Using the CNNUniversal Machine: New Analogic CNN Algorithms for Motion Compensation, Image Synthesis, and Consistency Observation" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, 2003.

[4] 4.VedatTav¸Sano˘glu" Decomposition of the Nodal Conductance Matrix ofa Planar Resistive Grid and Derivation of ItsEigenvalues and Eigenvectors Using the Kronecker Product and Sum WithApplication to CNN Image Filters" IEEE, DECEMBER ,2016.

[5] Pe´terFo¨ldesy, L. Ke´k, A´ kosZara´ndy" Fault-Tolerant Design of Analogic CNN Templatesand Algorithms—Part I: The Binary Output Case,IEEE,1999.

[6] Ce Zhang , Isabel Sargent, Xin Pan, Andy Gardiner, Jonathon Hare, and Peter M. Atkinson" VPRS-Based Regional Decision Fusion of CNN andMRF Classifications for Very Fine ResolutionRemotely Sensed Images"IEEE,2018.

[7] Mengmeng Zhang, Wei Li, Qian Du, Lianru Gao, Bing Zhang" Feature Extraction for Classification ofHyperspectral and LiDAR Data UsingPatch-to-Patch CNN",IEEE.

[8] JunpingZhong, ZhigangLiu ,Zhiwei Han Ye Han and Wenxuan Zhang" A CNN-Based Defect Inspection Method for Catenary Split Pins in High-Speed Railway",IEEE.

[9] Kun Niu1,Cheng Cheng1, Jielin Chang1, Huiyang Zhang1, Tong Zhou1" Real-Time Taxi-Passenger Prediction with L-CNN"

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in European Conference on Computer Vision (ECCV), 2014.

[11] R. Girshick, "Fast R-CNN," in IEEE International Conference on Computer Vision (ICCV), 2015.

ISSN: 2454-4248
79 – 85

[12] K. Simonyan and A. Zisserman, "Very deep convolutionanetworks for large-scale image recognition," in International Conference on Learning Representations (ICLR), 2015.
[13] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," International Journal of Computer Vision (IJCV), 2013.
[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[15] C. L. Zitnick and P. Dollar, "Edge boxes: Locating object ´ proposals from edges," in European Conference on Computer Vision (ECCV), 2014.
[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[17] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained partbased models," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2010.
[18] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in International Conference on Learning Representations (ICLR), 2014.
[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards 14 real-time object detection with region proposal networks," in Neural Information Processing Systems (NIPS), 2015.
[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," 2007.
[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Com- ´ mon Objects in Context," in European Conference on Computer Vision (ECCV), 2014.
[22] S. Song and J. Xiao, "Deep sliding shapes for amodal 3d object detection in rgb-d images," arXiv:1511.02300, 2015.
[23] J. Zhu, X. Chen, and A. L. Yuille, "DeePM: A deep part-based model for object detection and semantic part localization," arXiv:1511.07131, 2015.